IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application of Laurence B. Boucher et al.		Ser. No:	09/692,561
Filing Date:	October 18, 2000	Examiner:	Moustafa M. Meky
Atty. Docket No:	ALA-002A	GAU:	2157

For: INTELLIGENT NETWORK INTERFACE SYSTEM AND METHOD FOR ACCELERATED PROTOCOL PROCESSING

June 19, 2008

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Re-Submission of Previously Submitted Preliminary Amendment

Sir:

Accompanying this letter please find a copy of a Preliminary Amendment that was previously filed on June 12, 2002, for the above-identified application, along with proof that the Preliminary Amendment was filed that day. In a conversation between Examiner Meky and the undersigned on June 9, 2008, the Examiner stated that the Preliminary Amendment was shown in the electronic history in the Patent Office but the image of the Preliminary Amendment was not available, and the paper file could not be located. Similarly, the undersigned notes that a Preliminary Amendment dated June 12, 2002, is shown in the Transaction History portion of PAIR, but is not available in the Image File Wrapper portion of PAIR.

Also accompanying this letter is a copy of the Amendment Transmittal Letter that was submitted with the Preliminary Amendment on June 12, 2002. The Preliminary Amendment was filed by Express Mail, and a copy of the Express Mail label, stamped June 12, 2002, is also attached. Also submitted with this letter, for further proof of the Preliminary Amendment filing, are a copy of the Preliminary Amendment Postcard that was enclosed with the Preliminary Amendment, stamped as received by the U.S. Patent and Trademark Office on June 12, 2002, and a copy of the cancelled check, dated June 12, 2002, and paid by the undersigned's bank on June 24, 2002, in the amount of

$762.00, which was also enclosed with the Preliminary Amendment, for payment of additional claim fees.

Respectfully submitted,

CERTIFICATE OF ELECTRONIC TRANSMISSION
I hereby certify that this correspondence is being transmitted electronically to: the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on June 19, 2008.

Date: 6-19-08

Mark Lauer

Mark Lauer
Reg. No. 36,578
6601 Koll Center Parkway
Suite 245
Pleasanton, CA 94566
Tel: (925) 621-2121
Fax: (925) 621-2125

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants: Boucher, et al. Serial No.: 09/692,561

Filed: October 18, 2000 Examiner: Maung, Z.

Atty. Doc. No.: ALA-002A Group Art Unit: 2152

Assignee: Alacritech Inc.

Title: "Intelligent Network Interface System And Method For Accelerated Protocol Processing"

June 12, 2002

ASSISTANT COMMISSIONER FOR PATENTS
Washington, D.C. 20231

PRELIMINARY AMENDMENT

Sir:

Before examination on the merits, please amend the above-identified application as follows.

IN THE CLAIMS:

Please delete Claims 1 - 27, and add new Claims 28 - 59 as follows:

28. (New) A method for transferring data on a network from a data source to an end station executing a multi-layer network protocol, including a network layer and at least one higher layer, through a network interface on the end station, comprising:

receiving in the network interface a packet which carries a data payload from a block of data in the data source, and a control field identifying the packet;

determining based on the control field in the network interface whether the packet matches a flow specification, and if so transferring the data payload in the packet directly to a target buffer assigned by a process at a layer higher than the network layer.

29. (New) A method for transferring data on a network from a data source to an end station executing a multi-layer network protocol, including a network layer and at least one higher layer, through a network interface on the end station, comprising:

receiving in the network interface a packet which carries a data payload from a block of data in the data source, and a control field identifying the packet;

determining based on the control field in the network interface whether the packet matches a flow specification, and if so transferring the data payload in the packet directly to a target buffer assigned by a process at a layer higher than the network layer,

wherein the control field in the packet includes a packet header.

30. (New) A method for transferring data on a network from a data source to an end station executing a multi-layer network protocol, including a network layer and at least one higher layer, through a network interface on the end station, comprising:

receiving in the network interface a packet which carries a data payload from a block of data in the data source, and a control field identifying the packet;

determining based on the control field in the network interface whether the packet matches a flow specification, and if so transferring the data payload in the packet directly to a target buffer assigned by a process at a layer higher than the network layer,

wherein the multi-layer network protocol comprises TCP/IP, and the control field comprises a TCP/IP header.

31. (New) A method for transferring data on a network from a data source to an end station executing a multi-layer network protocol, including a network layer and at least one higher layer, through a network interface on the end station, comprising:

receiving in the network interface a packet which carries a data payload from a block of data in the data source, and a control field identifying the packet;

determining based on the control field in the network interface whether the packet matches a flow specification, and if so transferring the data payload in the packet directly to a target buffer assigned by a process at a layer higher than the network layer,

including prior to receiving the packet, allocating the target buffer for the plurality of packets, and notifying the network interface of the allocated target buffer.

32. (New) A method for transferring data on a network from a data source to an end station executing a multi-layer network protocol, including a network layer and at least one higher layer, through a network interface on the end station, comprising:

receiving in the network interface a packet which carries a data payload from a block of data in the data source, and a control field identifying the packet;

determining based on the control field in the network interface whether the packet matches a flow specification, and if so transferring the data payload in the packet directly to a target buffer assigned by a process at a layer higher than the network layer,

the network interface is coupled to a network medium supporting a maximum packet size, and including transmitting a request from an application for transfer of a block of data from the data source, the block of

data having a length potentially greater than the maximum packet size for the medium.

33. (New) A method for transferring data on a network from a data source to an end station executing a multi-layer network protocol, including a network layer and at least one higher layer, through a network interface on the end station, comprising:

receiving in the network interface a packet which carries a data payload from a block of data in the data source, and a control field identifying the packet;

determining based on the control field in the network interface whether the packet matches a flow specification, and if so transferring the data payload in the packet directly to a target buffer assigned by a process at a layer higher than the network layer, the network interface is coupled to a network medium supporting a maximum packet size, and including transmitting a request from an application for transfer of a block of data from the data source, the block of data having a length potentially greater than the maximum packet size for the medium,

including notifying the network interface in response to the request of a flow specification for the block of data according to the multi-layer network protocol, and wherein the step of receiving the packet includes identifying packet using the flow specification.

34. (New) A method for transferring data on a network from a data source to an end station executing a multi-layer network protocol, including a network layer and at least one higher layer, through a network interface on the end station, comprising:

receiving in the network interface a packet which carries a data payload from a block of data in the data source, and a control field identifying the packet;

determining based on the control field in the network interface whether the packet matches a flow specification, and if so transferring the data payload in the packet directly to a target buffer assigned by a process at a layer higher than the network layer, the network interface is coupled to a network medium supporting a maximum packet size, and including transmitting a request from an application for transfer of a block of data from the data source, the block of data having a length potentially greater than the maximum packet size for the medium,

including notifying the network interface in response to the request of a flow specification for the block of data according to the multi-layer network protocol, and wherein the step of receiving the packet includes identifying packet using the flow specification,

wherein the network protocol comprises TCP/IP, and the flow specification includes a sequence number of a first byte from the plurality of packets to be stored in the target buffer.

35. (New) A method for transferring data on a network from a data source to an end station executing a multi-layer network protocol, including a network layer and at least one higher layer, through a network interface on the end station, comprising:

receiving in the network interface a packet which carries a data payload from a block of data in the data source, and a control field identifying the packet;

determining based on the control field in the network interface whether the packet matches a flow specification, and if so transferring the

data payload in the packet directly to a target buffer assigned by a process at a layer higher than the network layer,

wherein the flow specification includes a sequence number for the block of data.

36. (New) A method for transferring data on a network from a data source to an end station executing a multi-layer network protocol, including a network layer and at least one higher layer, through a network interface on the end station, comprising:

receiving in the network interface a packet which carries a data payload from a block of data in the data source, and a control field identifying the packet;

determining based on the control field in the network interface whether the packet matches a flow specification, and if so transferring the data payload in the packet directly to a target buffer assigned by a process at a layer higher than the network layer,

wherein the flow specification includes a sequence number for the block of data, and the flow specification includes IP source and destination addresses and TCP port numbers.

37. (New) The method of claim 28, wherein control of the flow specification is transferred from the network interface so that a second packet that matches the flow specification is processed by the network layer.

38. (New) The method of claim 28, wherein determining based on the control field in the network interface whether the packet matches a flow specification is performed by specialized network processing hardware.

39. (New) The method of claim 28, wherein the target buffer is assigned by processing an initial packet with the process.

40. (New) The method of claim 28, wherein the network interface comprises means for determining whether the packet matches the flow specification.

41. (New) A method for transferring data on a network from a data source to an end station executing a multi-layer network protocol, including a network layer and at least one higher layer, through a network interface on the end station, comprising:

receiving in the network interface a packet which carries a data payload from a block of data in the data source, and a control field identifying the packet;

determining based on the control field in the network interface whether the packet matches a transmit control block (TCB), and if so transferring the data payload in the packet directly to a target buffer assigned by a process at a layer higher than the network layer.

42. (New) The method of claim 41, wherein the control field in the packet includes a packet header.

43. (New) The method of claim 41, wherein the multi-layer network protocol comprises TCP/IP, and the control field comprises a TCP/IP header.

44. (New) The method of claim 41, including prior to receiving the packet, allocating the target buffer for a plurality of packets, and notifying the network interface of the allocated target buffer.

45. (New) The method of claim 41, wherein the network interface is coupled to a network medium supporting a maximum packet size, and including transmitting a request from an application for transfer of a block of data from the data source, the block of data having a length potentially greater than the maximum packet size for the medium.

46. (New) The method of claim 45, including notifying the network interface in response to the request of a TCB for the block of data according to the multi-layer network protocol, and wherein the step of receiving the packet includes identifying packet using the TCB.

47. (New) The method of claim 46, wherein the network protocol comprises TCP/IP, and the TCB includes a sequence number of a first byte from the plurality of packets to be stored in the target buffer.

48. (New) The method of claim 41, wherein the TCB includes a sequence number for the block of data.

49. (New) The method of claim 48, wherein the TCB includes IP source and destination addresses and TCP port numbers.

50. (New) The method of claim 41, wherein control of the TCB is transferred from the network interface so that a second packet that matches the TCB is processed by the network layer.

51. (New) The method of claim 41, wherein determining based on the control field in the network interface whether the packet matches the TCB is performed by an application specific integrated circuit (ASIC).

52. (New) The method of claim 41, wherein the target buffer is assigned by processing an initial packet with the process.

53. (New) The method of claim 41, wherein the target buffer is identified with a memory descriptor list (MDL).

54. (New) A method for transferring data on a network from a data source to an end station, the end station comprising a host portion and a network interface portion, the host portion executing a multi-layer network protocol including a network layer and at least one higher layer, the data being transferred through the network interface portion of the end station, comprising:

transferring from the host portion to the network interface portion a connection definition, the connection definition comprising an IP source address, an IP destination address, a TCP source port and a TCP destination port;

receiving onto the network interface portion a first packet of a multi-packet message, the multi-packet message being a message of a layer higher than the network layer, the first packet including a control field identifying the first packet, the multi-packet message having a data payload;

reading onto the network interface portion a command from a command buffer, the command including an identifier of a target buffer, the target buffer having been assigned on the host portion by a process at a layer higher than the network layer;

determining based on the control field in the network interface portion whether the first packet corresponds to the connection definition,

and if so transferring the data payload of the multi-packet message from the network interface portion directly to the target buffer; and

the network interface portion writing to a response buffer and thereby indicating to the host portion completion of the command.

55. (New) The method of Claim 50, wherein the network interface portion writes status information and a command buffer identifier into the response buffer.

56. (New) The method of Claim 50, wherein the identifier of the target buffer is a list of addresses.

57. (New) The method of Claim 50, wherein the identifier of the target buffer is a list of data pointers.

58. (New) The method of Claim 50, wherein the connection definition is a part of a Transmit Control Block (TCB).

59. (New) The method of Claim 50, wherein the data payload is transferred to the host portion after the headers of the packets of the multi-packet message have been stripped.

REMARKS

Applicants hereby seek to have an interference declared between the present application (09/692,561) and U.S. Patent No. 6,246,683, issued June 12, 2001.

U.S. Patent No. 6,246,683 (Connery, et al.), issued from U.S. application serial number 09/071,692 (filed May 1, 1998).

Applicants' present application (U.S. patent application serial no. 09/692,561, filed October 18, 2000) claims priority from, and incorporates by reference, prior U.S. patent application serial number 09/067,544 (filed April 27, 1998). Prior U.S. patent application serial number 09/067,544 in turn incorporates by reference and claims the benefit under 35 U.S.C. §119 of provisional application number 60/061,809 (filed October 14, 1997), which is also incorporated by reference in the present application.

Claims 28-36 of the present application are copied from Claims 1-9 of U.S. Patent No. 6,246,683, respectively.

REQUIREMENTS OF 37 CFR 607:

Per rule 607(a)(1), the patent is U.S. Patent No. 6,246,683.

Per rule 607(a)(2), Applicants submit nine proposed counts as set forth below.

Count #1:

A method for transferring data on a network from a data source to an end station executing a multi-layer network protocol, including a network layer and at least one higher layer, through a network interface on the end station, comprising:

receiving in the network interface a packet which carries a data payload from a block of data in the data source, and a control field identifying the packet;

determining based on the control field in the network interface whether the packet matches a flow specification, and if so transferring the data payload in the packet directly to a target buffer assigned by a process at a layer higher than the network layer.

Count #2:

A method for transferring data on a network from a data source to an end station executing a multi-layer network protocol, including a network layer and at least one higher layer, through a network interface on the end station, comprising:

receiving in the network interface a packet which carries a data payload from a block of data in the data source, and a control field identifying the packet;

determining based on the control field in the network interface whether the packet matches a flow specification, and if so transferring the data payload in the packet directly to a target buffer assigned by a process at a layer higher than the network layer,

wherein the control field in the packet includes a packet header.

Count #3:

A method for transferring data on a network from a data source to an end station executing a multi-layer network protocol, including a network layer and at least one higher layer, through a network interface on the end station, comprising:

receiving in the network interface a packet which carries a data payload from a block of data in the data source, and a control field identifying the packet;

determining based on the control field in the network interface whether the packet matches a flow specification, and if so transferring the data payload in the packet directly to a target buffer assigned by a process at a layer higher than the network layer,

wherein the multi-layer network protocol comprises TCP/IP, and the control field comprises a TCP/IP header.

Count #4:

A method for transferring data on a network from a data source to an end station executing a multi-layer network protocol, including a network layer and at least one higher layer, through a network interface on the end station, comprising:

receiving in the network interface a packet which carries a data payload from a block of data in the data source, and a control field identifying the packet;

determining based on the control field in the network interface whether the packet matches a flow specification, and if so transferring the data payload in the packet directly to a target buffer assigned by a process at a layer higher than the network layer,

including prior to receiving the packet, allocating the target buffer for the plurality of packets, and notifying the network interface of the allocated target buffer.

Count #5:

A method for transferring data on a network from a data source to an end station executing a multi-layer network protocol, including a network layer and at least one higher layer, through a network interface on the end station, comprising:

receiving in the network interface a packet which carries a data payload from a block of data in the data source, and a control field identifying the packet;

determining based on the control field in the network interface whether the packet matches a flow specification, and if so transferring the

data payload in the packet directly to a target buffer assigned by a process at a layer higher than the network layer,

the network interface is coupled to a network medium supporting a maximum packet size, and including transmitting a request from an application for transfer of a block of data from the data source, the block of data having a length potentially greater than the maximum packet size for the medium.

Count #6:

A method for transferring data on a network from a data source to an end station executing a multi-layer network protocol, including a network layer and at least one higher layer, through a network interface on the end station, comprising:

receiving in the network interface a packet which carries a data payload from a block of data in the data source, and a control field identifying the packet;

determining based on the control field in the network interface whether the packet matches a flow specification, and if so transferring the data payload in the packet directly to a target buffer assigned by a process at a layer higher than the network layer,

the network interface is coupled to a network medium supporting a maximum packet size, and including transmitting a request from an application for transfer of a block of data from the data source, the block of data having a length potentially greater than the maximum packet size for the medium,

including notifying the network interface in response to the request of a flow specification for the block of data according to the multi-layer network protocol, and wherein the step of receiving the packet includes identifying packet using the flow specification.

Count #7:

A method for transferring data on a network from a data source to an end station executing a multi-layer network protocol, including a network layer and at least one higher layer, through a network interface on the end station, comprising:

receiving in the network interface a packet which carries a data payload from a block of data in the data source, and a control field identifying the packet;

determining based on the control field in the network interface whether the packet matches a flow specification, and if so transferring the data payload in the packet directly to a target buffer assigned by a process at a layer higher than the network layer,

the network interface is coupled to a network medium supporting a maximum packet size, and including transmitting a request from an application for transfer of a block of data from the data source, the block of data having a length potentially greater than the maximum packet size for the medium,

including notifying the network interface in response to the request of a flow specification for the block of data according to the multi-layer network protocol, and wherein the step of receiving the packet includes identifying packet using the flow specification,

wherein the network protocol comprises TCP/IP, and the flow specification includes a sequence number of a first byte from the plurality of packets to be stored in the target buffer.

Count #8:

A method for transferring data on a network from a data source to an end station executing a multi-layer network protocol, including a network

layer and at least one higher layer, through a network interface on the end station, comprising:

receiving in the network interface a packet which carries a data payload from a block of data in the data source, and a control field identifying the packet;

determining based on the control field in the network interface whether the packet matches a flow specification, and if so transferring the data payload in the packet directly to a target buffer assigned by a process at a layer higher than the network layer,

wherein the flow specification includes a sequence number for the block of data.

Count #9:

A method for transferring data on a network from a data source to an end station executing a multi-layer network protocol, including a network layer and at least one higher layer, through a network interface on the end station, comprising:

receiving in the network interface a packet which carries a data payload from a block of data in the data source, and a control field identifying the packet;

determining based on the control field in the network interface whether the packet matches a flow specification, and if so transferring the data payload in the packet directly to a target buffer assigned by a process at a layer higher than the network layer,

wherein the flow specification includes a sequence number for the block of data, and the flow specification includes IP source and destination addresses and TCP port numbers.

Per rule 607(a)(3), Claim 1 in the patent (6,246,683) corresponds to Count #1.

Per rule 607(a)(4), Claim 28 in the application (09/692,561) corresponds exactly to Count #1.

Per rule 607(a)(3), Claim 2 in the patent (6,246,683) corresponds to Count #2.

Per rule 607(a)(4), Claim 29 in the application (09/692,561) corresponds exactly to Count #2.

Per rule 607(a)(3), Claim 3 in the patent (6,246,683) corresponds to Count #3.

Per rule 607(a)(4), Claim 30 in the application (09/692,561) corresponds exactly to Count #3.

Per rule 607(a)(3), Claim 4 in the patent (6,246,683) corresponds to Count #4.

Per rule 607(a)(4), Claim 31 in the application (09/692,561) corresponds exactly to Count #4.

Per rule 607(a)(3), Claim 5 in the patent (6,246,683) corresponds to Count #5.

Per rule 607(a)(4), Claim 32 in the application (09/692,561) corresponds exactly to Count #5.

Per rule 607(a)(3), Claim 6 in the patent (6,246,683) corresponds to Count #6.

Per rule 607(a)(4), Claim 33 in the application (09/692,561) corresponds exactly to Count #6.

Per rule 607(a)(3), Claim 7 in the patent (6,246,683) corresponds to Count #7.

Per rule 607(a)(4), Claim 34 in the application (09/692,561) corresponds exactly to Count #7.

Per rule 607(a)(3), Claim 8 in the patent (6,246,683) corresponds to Count #8.

Per rule 607(a)(4), Claim 35 in the application (09/692,561) corresponds exactly to Count #8.

Per rule 607(a)(3), Claim 9 in the patent (6,246,683) corresponds to Count #9.

Per rule 607(a)(4), Claim 36 in the application (09/692,561) corresponds exactly to Count #9.

Per rule 607(a)(5), application of the terms of Claims 28-36 to the disclosure of the above-identified application is set forth in the claims chart below. In the claim chart below, [A1] designates the present application 09/692,561, [A2] designates the parent application 09/067,544, and [A3] designates the provisional application 60/061,809.

Claims copied from USP 6,246,683.	Disclosure in Applicants' application (including Applicants' prior applications that are incorporated by reference).
28. (New) A method for transferring data on a network from a data source to an end station executing a multi-layer network protocol, including a network layer and at least one higher layer, through a network interface on the end station, comprising:	[A2, page 7, line 8 through page 8, line 8] Figures 1-8 relate to a "first embodiment". [A2, page 9, Fig. 1, and lines 2-8] In Fig. 1, remote host 22 is a data source; and host 20 is an end station. Host 20 includes a CPU 28 and a CPD 30. As illustrated, CPD 30 interfaces host 20 to network 25. [A2, page 9, lines 13-15] CPU 28 of host 20 executes a protocol processing "stack" 44. The "stack 44" includes a "network layer" 38 and a "transport layer" 40. [A2, FIG. 1] CPD 30 interfaces host 20 to network 25.
receiving in the network interface a packet which carries a data payload from a block of data in the data source, and a control field identifying the packet;	[A2, page 11, lines 14-18] A "TCP/IP message" is "received by the host from the network" in the form of many separate "frames" or "packets" (an initial packet, and subsequent packets). [A2, page 4, line 20-23] "Each layer of the receiving host recognizes and manipulates only the headers associated with that layer, since to that layer the higher layer control data is included with and indistinguishable from the payload data." [A2, page 15, lines 4-6] One of the "subsequent packets" is received from network 25 by CPD 30. This packet includes a "packet header" and "data". [A2, page 12, lines 1-3] "...each packet conventionally includes a portion of data being transferred, as well as headers for each of the protocol layers and markers for positioning the packet relative to the rest of the packets of this message".
determining based on the control field in the network interface whether the packet matches a flow specification, and if so transferring the data payload in the packet directly to a target buffer assigned by a process at a layer higher than the network layer.	[A2, page 15, lines 4-12] The packet "header" of the subsequent packet is "parsed to create a summary of the message packet and a hash for finding a corresponding CCB..." The term "flow specification" corresponds to the term "CCB" as the term is used in [A2]. [A2, page 24, lines 13-18] "A CCB includes connection and state information regarding the protocol layers and packets of the message. Thus a CCB can include source and destination media access control (MAC) addresses, source and destination IP or IPX addresses, source and destination TCP or SPX ports, TCP variables such as timers, receive and transmit windows for sliding window protocols, and information denoting the session layer protocol". [A2, page 15, lines 9-12] " The processor 55 checks for a ***match*** between the hash and each CCB that is stored in the cache 62 and, finding a match, sends the ***data*** (D2) 70 via a fast-path ***directly to the destination*** in storage 35,..." (emphasis added). [A2, page 12, line 21 through page 13, line 5] "All received message frames which have been determined by the CPD hardware assist to be fast-path candidates are examined 53 by the network microprocessor on INIC comparitor circuits to ***determine*** whether they ***match*** a CCB held by the CPD. Upon confirming such a match, the CPD removes lower layer headers and sends 69 the remaining application ***data*** from the frame ***directly into its final destination in the host*** using direct memory access (DMA) units of the CPD. This operation may occur immediately upon receipt of a message packet, for example when a TCP connection already exists..." (emphasis added).

	[A2, page 18, lines 2-3] The "***application*** layer 166...***provides a*** source or ***destination*** 168 for the communication data..."
29. (New) A method for transferring data on a network from a data source to an end station executing a multi-layer network protocol, including a network layer and at least one higher layer, through a network interface on the end station, comprising: receiving in the network interface a packet which carries a data payload from a block of data in the data source, and a control field identifying the packet; determining based on the control field in the network interface whether the packet matches a flow specification, and if so transferring the data payload in the packet directly to a target buffer assigned by a process at a layer higher than the network layer, wherein the control field includes a packet header.	See explanation above regarding claim 28. [A2, page 13, line 1] "Upon confirming such a match, the CPD removes lower layer ***headers*** and sends 69 the remaining application data from the frame directly into its final destination in the host ..."
30. (New) A method for transferring data on a network from a data source to an end station executing a multi-layer network protocol, including a network layer and at least one higher layer, through a network interface on the end station, comprising: receiving in the network interface a packet which carries a data payload from a block of data in the data source, and a control field identifying the packet; determining based on the control field in the network interface whether the packet matches a flow specification, and if so transferring the data payload in the packet directly to a target buffer assigned by a process at a layer higher than the network layer, wherein the multi-layer network protocol comprises TCP/IP, and the control field comprises a TCP/IP header.	See explanation above regarding claim 28. [A2, page 11, line 15] The message being received by the host is a "TCP/IP message". [A2, page 4, line 20-23] "Each layer of the receiving host recognizes and manipulates only the headers associated with that layer, since to that layer the higher layer control data is included with and indistinguishable from the payload data." [A2, page 12, lines 1-3] "...each packet conventionally includes a portion of the data being transferred, as well as headers for each of the protocol layers..." [A2, page 12, lines 9-12] "Selection of fast-path candidates is based on whether the host may benefit from this message connection being handled by the CPD, which includes determining whether the packet has header bytes denoting particular protocols, such as TCP/IP..."

31. (New) A method for transferring data on a network from a data source to an end station executing a multi-layer network protocol, including a network layer and at least one higher layer, through a network interface on the end station, comprising: receiving in the network interface a packet which carries a data payload from a block of data in the data source, and a control field identifying the packet; determining based on the control field in the network interface whether the packet matches a flow specification, and if so transferring the data payload in the packet directly to a target buffer assigned by a process at a layer higher than the network layer,	See explanation above regarding claim 28.
including prior to receiving the packet, allocating the target buffer for the plurality of packets,	[A2, page 15, lines 1-5] Prior to receiving the "subsequent packet" as set forth on page 15, the "initial packet" is received. [A2, page 14, lines 16-28] The host uses the "initial packet" to "to create a connection context for the message, including finding and reserving a ***destination*** for the data from the message associated with the packet, the context taking the form of a CCB..."
and notifying the network interface of the allocated target buffer.	[A2, page 14, line 18] "The CCB is then sent to the CPD 30 to be saved in cache 62...".
32. (New) A method for transferring data on a network from a data source to an end station executing a multi-layer network protocol, including a network layer and at least one higher layer, through a network interface on the end station, comprising: receiving in the network interface a packet which carries a data payload from a block of data in the data source, and a control field identifying the packet; determining based on the control field in the network interface whether the packet matches a flow specification, and if so transferring the data payload in the packet directly to a target buffer assigned by a process at a layer higher than the network layer,	See explanation above regarding claim 28.
the network interface is coupled to a network medium supporting a maximum packet size,	[A2, Fig. 12] INIC 150 of Fig. 12 is a "network interface" of "client" 190. [A2, page 11, lines 14-18] "A large TCP/IP message ... may be received by the host ...in a number of separate, approximately 64 KB transfers, each of which may be split into many, approximately 1.5 KB frames or packets for transmission over a network."
and including transmitting a request from an application for transfer of a block of data from the data source,	[A2, page 29, line 20 through page 30, line 14] SMB is an application. An SMB read "request" to read a "100KB file" is transmitted from INIC 150 to server 290. Server 290 (INIC 200 of server 290) sends the 100 KB file back to INIC 150 as multiple "packets".

the block of data having a length potentially greater than the maximum packet size for the medium.	It is well known in networking that a network, such as network 25, has a maximum packet size. See, for example, the discussion of "maximum packet size" in the text incorporated by reference into [A2]: Computer Networks, Third Edition (1996) by Andrew Tanenbaum, page 406-409.
33. (New) A method for transferring data on a network from a data source to an end station executing a multi-layer network protocol, including a network layer and at least one higher layer, through a network interface on the end station, comprising: receiving in the network interface a packet which carries a data payload from a block of data in the data source, and a control field identifying the packet; determining based on the control field in the network interface whether the packet matches a flow specification, and if so transferring the data payload in the packet directly to a target buffer assigned by a process at a layer higher than the network layer, the network interface is coupled to a network medium supporting a maximum packet size, and including transmitting a request from an application for transfer of a block of data from the data source, the block of data having a length potentially greater than the maximum packet size for the medium,	See explanation above regarding claim 32.
including notifying the network interface in response to the request of a flow specification for the block of data according to the multi-layer network protocol,	A2, page 14, line 14 through page 15, line 3] The first packet received is sent "to the host protocol stack 44 for processing. Host stack 44 may use this packet to create a connection context for the message...the context taking the form of a CCB...The CCB is then sent to the CPD 30 to be saved in cache 62."
and wherein the step of receiving the packet includes identifying packet using the flow specification.	[A2, page 15, lines 4-14] When "a subsequent packet from the same connection as the initial packet" is received, the receiving INIC checks "for a match between the hash and each CCB that is stored in the cache 62 and, finding a match..." [A2, page 24, lines 7-9] "A TCP/IP...message has a connection that is set up from which a CCB is formed by the driver and passed to the INIC for matching with and guiding the fast-path packet to the connection destination 168".
34. (New) A method for transferring data on a network from a data source to an end station executing a multi-layer network protocol, including a network layer and at least one higher layer, through a network interface on the end station, comprising: receiving in the network interface a packet which carries a data payload from	See explanation above regarding claim 33.

a block of data in the data source, and a control field identifying the packet; determining based on the control field in the network interface whether the packet matches a flow specification, and if so transferring the data payload in the packet directly to a target buffer assigned by a process at a layer higher than the network layer, including notifying the network interface in response to the request of a flow specification for the block of data according to the multi-layer network protocol, the network interface is coupled to a network medium supporting a maximum packet size, and including transmitting a request from an application for transfer of a block of data from the data source, the block of data having a length potentially greater than the maximum packet size for the medium, including notifying the network interface in response to the request of a flow specification for the block of data according to the multi-layer network protocol, and wherein the step of receiving the packet includes identifying packet using the flow specification, wherein the network protocol comprises TCP/IP, and the flow specification includes a sequence number of a first byte from the plurality of packets to be stored in the target buffer.	The term "flow specification" corresponds to the term "TCB" as the term is used in [A3]. [A3, page 6, lines 18-25] "A TCB is a structure that contains the entire context associated with a connection. This includes the source and destination IP addresses and source and destination TCP ports that define the connection. It also contains information about the connection itself such as the current send and receive ***sequence numbers***, and the first-hop MAC address, etc."
35. (New) A method for transferring data on a network from a data source to an end station executing a multi-layer network protocol, including a network layer and at least one higher layer, through a network interface on the end station, comprising: receiving in the network interface a packet which carries a data payload from a block of data in the data source, and a control field identifying the packet; determining based on the control field in the network interface whether the packet matches a flow specification, and if so transferring the data payload in the packet directly to a target buffer assigned by a process at a layer higher than the network layer,	See explanation above regarding claim 28.

wherein the flow specification includes a sequence number for the block of data.	The term "flow specification" corresponds to the term "CCB" as the term is used in [A2]. [A2, page 24, lines 13-18] "A CCB includes connection and state information regarding the protocol layers and packets of the message. Thus a CCB can include source and destination media access control (MAC) addresses, source and destination IP or IPX addresses, source and destination TCP or SPX ports, TCP variables such as timers, receive and transmit windows for sliding window protocols, and information denoting the session layer protocol". Tanenbaum, (incorporated by reference in A2) page 203 lines 28-35. "In all sliding window protocols, each outbound frame contains a sequence number, ...The essence of all sliding window protocols is that at any instant of time, the sender maintains a set of sequence numbers corresponding to frames it is permitted to send."
36. (New) A method for transferring data on a network from a data source to an end station executing a multi-layer network protocol, including a network layer and at least one higher layer, through a network interface on the end station, comprising: receiving in the network interface a packet which carries a data payload from a block of data in the data source, and a control field identifying the packet; determining based on the control field in the network interface whether the packet matches a flow specification, and if so transferring the data payload in the packet directly to a target buffer assigned by a process at a layer higher than the network layer, wherein the flow specification includes a sequence number for the block of data, and the flow specification includes IP source and destination addresses and TCP port numbers.	See explanation above regarding claim 35. [A2, page 24, lines 13-18] "A CCB includes connection and state information regarding the protocol layers and packets of the message. Thus a CCB can include source and destination media access control (MAC) addresses, source and destination IP or IPX addresses, source and destination TCP or SPX ports, TCP variables such as timers, receive and transmit windows for sliding window protocols, and information denoting the session layer protocol".
Other Claims	Disclosure in Applicants' application (including Applicants' prior applications that are incorporated by reference).
37. (New) The method of claim 28, wherein control of the flow specification is transferred from the network interface so that a second packet that matches the flow specification is processed by the network layer.	[A2, page 15, lines 15-22] "FIG. 4D shows the procedure for handling the rare instance when a message for which a fast-path connection has been established, such as shown in FIG. 4C, has a packet that is not easily handled by the CPD. In this case the packet is sent to be processed by the protocol stack 44, which is handed the CCB for that message from cache 62 via a control dialogue with the CPD, as shown by arrow 76, signaling to the CPU to take over processing of that message. Slow-path processing by the protocol stack then results in data (D3) 80 from the packet being sent, as shown by arrow 82, to storage 35."
38. (New) The method of claim 28, wherein determining based on the control field in the network interface whether the	[A2, page 15, lines 9-12] "The processor 55 checks for a match between the hash and each CCB that is stored in the cache 62 and, finding a match, sends the data (D2) 70 via a fast-path directly to the destination in storage 35, as shown by arrow 72, bypassing the

packet matches a flow specification is performed by specialized network processing hardware.	session layer 42, transport layer 40, network layer 38 and data link layer 36."
39. (New) The method of claim 28, wherein the target buffer is assigned by processing an initial packet with the process.	[A2, page 14, lines 16-18] "Host stack 44 may use this packet to create a connection context for the message, including finding and reserving a destination for data from the message associated with the packet, the context taking the form of a CCB."
40. (New) The method of claim 28, wherein the network interface comprises means for determining whether the packet matches the flow specification.	For example, see A2, FIG. 1, FIG. 7, FIG. 8 and FIG. 13 and corresponding text.
41. (New) A method for transferring data on a network from a data source to an end station	[A3, page 2, lines 36-37] "A 64k SMB request (write or read-reply) is typically made up of 44 TCP segments when running over Ethernet" [A3, page 3, lines 34-35] "Alacritech was formed with the idea that the network processing described above could be offloaded onto a cost-effective Intelligent Network Interface Card (INIC)." [A3, page 6] The figure shows an end station.
executing a multi-layer network protocol, including a network layer and at least one higher layer,	[A3, page 6] The figure shows a multi-layer network protocol, including a network layer and at least one higher layer. [A3, page 18, lines 22-26] "This section outlines the design specification for the Alacritech TCP (ATCP) transport driver. The ATCP driver consists of three components: 1. The bulk of the protocol stack is based on the FreeBSD TCP/IP protocol stack. This code performs the Ethernet, ARP, IP, ICMP, and (slow path) TCP processing for the driver..."
through a network interface on the end station, comprising:	[A3, page 6] The figure shows the INIC interfacing the end station to the Ethernet network.
receiving in the network interface a packet which carries a data payload from a block of data in the data source, and a control field identifying the packet;	[A3, page 2, lines 36-37] "A 64k SMB request (write or read-reply) is typically made up of 44 TCP segments when running over Ethernet" [USP 6,246,683, column 2, lines 59-65] "the process for requesting the transfer of a file from a data source involves issuing a read request according to higher layer protocol, such as the READ RAW SMB (server message block) command specified according to the Common Internet File System protocol (See, paragraph 3.9.35 of CIFS/1.0 draft dated Jun. 13, 1996) executed in Windows platforms." [A3, page 2, lines 29-30] " The TCP connection object must be located when a given TCP segment arrives, IP header checksums must be calculated..."
determining based on the control field in the network interface	[A3, page 7, lines 13-16] "When a frame is received by the INIC, it must verify it completely before it even determines whether it belongs to one of its TCBs or not. This includes all header validation (is it IP, IPV4 or V6, is the IP header checksum correct, is the TCP checksum correct, etc)."
whether the packet matches a transmit control block (TCB),	[A3, page 7, lines 16-18] "Once this is done it must compare the source and destination IP address and the source and destination TCP port with those in each of its TCBs to determine if it is associated with one of its TCBs."
and if so transferring the data payload in the packet directly to a target buffer assigned by a process at a layer higher than the network layer.	[A3, page 3, lines 38-39] "The vast majority of the data is moved directly from the INIC into its final destination." [A3, page 8, lines 9-12] "When this small amount of data is passed up to the client, and it returns with the address in which to put the remainder of the data, our host transport driver will pass that address to the INIC which will DMA the remainder of the data into its final destination."

42. (New) The method of Claim 41, wherein the control field is a packet header.	[A3, page 7, lines 13-14] "When a frame is received by the INIC, it must verify it completely before it even determines whether it belongs to one of its TCBs or not. This includes all header validation (is it IP, IPV4 or V6, is the IP header checksum correct, is the TCP checksum correct, etc)."
43. (New) The method of claim 41, wherein the multi-layer network protocol comprises TCP/IP, and the control field comprises a TCP/IP header.	[A3, page 18, lines 22-26] "This section outlines the design specification for the Alacritech TCP (ATCP) transport driver. The ATCP driver consists of three components: 1. The bulk of the protocol stack is based on the FreeBSD TCP/IP protocol stack. This code performs the Ethernet, ARP, IP, ICMP, and (slow path) TCP processing for the driver..." [A3, page 7, lines 13-14] "When a frame is received by the INIC, it must verify it completely before it even determines whether it belongs to one of its TCBs or not. This includes all header validation (is it IP, IPV4 or V6, is the IP header checksum correct, is the TCP checksum correct, etc)."
44. (New) The method of claim 41, including prior to receiving the packet, allocating the target buffer for a plurality of packets, and notifying the network interface of the allocated target buffer.	[A3, page 8, lines 9-12] "When this small amount of data is passed up to the client, and it returns with the address in which to put the remainder of the data, our host transport driver will pass that address to the INIC which will DMA the remainder of the data into its final destination." [A3, page 21, lines 14-47] "As soon as the INIC has received a segment containing a NETBIOS header, it will forward it up to the TCP driver, along with the NETBIOS length from the header.... On receiving the indicated packet, the ATCP driver will call the receive handler registered by the TDI client for the connection, passing the actual size of the data in the packet from the INIC as "bytes indicated" and the NETBIOS length as "bytes available"... In the "large data input" case, where "bytes available" exceeds the packet length, the TDI client will then provide an MDL... The ATCP driver will build a "receive request" from the MDL information, and pass this to the INIC. This request will contain:... A list of physical addresses corresponding to the MDL pages."
45. (New) The method of claim 41, wherein the network interface is coupled to a network medium supporting a maximum packet size, and including transmitting a request from an application for transfer of a block of data from the data source, the block of data having a length potentially greater than the maximum packet size for the medium.	[A3, page 6] The figure shows the INIC coupled to the Ethernet network. [A3, page 2, line 37] "Ethernet (1500 byte MTU)." [A3, page 2, lines 36-37] "A 64k SMB request (write or read-reply) is typically made up of 44 TCP segments when running over Ethernet (1500 byte MTU)." [A3, page 4, line 37] "a single 64k SMB write is broken down into 44 1500 byte TCP segments, which are in turn broken down into 131 576 byte IP fragments."
46. (New) The method of claim 45, including notifying the network interface in response to the request of the TCB for the block of data according to the multi-layer network protocol, and wherein the step of receiving the packet includes identifying packet using the flow specification.	[A2, page 14, line 14 through page 15, line 3] The first packet received is sent "to the host protocol stack 44 for processing. Host stack 44 may use this packet to create a connection context for the message...the context taking the form of a CCB...The CCB is then sent to the CPD 30 to be saved in cache 62." [A2, page 15, lines 4-14] When "a subsequent packet from the same connection as the initial packet" is received, the receiving INIC checks "for a match between the hash and each CCB that is stored in the cache 62 and, finding a match..." [A2, page 24, lines 7-9] "A TCP/IP...message has a connection that is set up from which a CCB is formed by the driver and passed to the INIC for matching with and guiding the fast-path packet to the connection destination 168".

47. (New) The method of claim 46, wherein the network protocol comprises TCP/IP, and the TCB includes a sequence number from the plurality of packets to be stored in the target buffer.	[A3, page 6, lines 38-41] "A TCB is a structure that contains the entire context associated with a connection. This includes the source and destination IP addresses and source and destination TCP ports that define the connection. It also contains information about the connection itself such as the current send and receive sequence numbers."
48. (New) The method of claim 41, wherein the TCB includes a sequence number for the block of data.	[A3, page 6, lines 38-41] "A TCB is a structure that contains the entire context associated with a connection. This includes the source and destination IP addresses and source and destination TCP ports that define the connection. It also contains information about the connection itself such as the current send and receive sequence numbers."
49. (New) The method of claim 48, wherein the TCB includes IP source and destination addresses and TCP port numbers.	[A3, page 6, lines 38-41] "A TCB is a structure that contains the entire context associated with a connection. This includes the source and destination IP addresses and source and destination TCP ports that define the connection. It also contains information about the connection itself such as the current send and receive sequence numbers."
50. (New) The method of claim 41, wherein control of the TCB is transferred from the network interface so that a second packet that matches the TCB is processed by the network layer.	[A3, page 6, line 28– page 7, line 8] "From INIC to ATCP...a context transfer may be initiated either by the ATCP driver or by the INIC...If the ATCP driver wishes to cause context to be flushed from INIC to host, it will send a "flush" message to the INIC specifying the context number to be flushed. Once the INIC receives this, it will proceed with the same steps as for the case where the flush is initiated by the INIC itself... At this point, the INIC is no longer doing fast-path processing, and any further incoming frames for the connection will simply be sent to the host as raw frames for the slow input path."
51. (New) The method of claim 41, wherein determining based on the control field in the network interface whether the packet matches the TCB is performed by an application specific integrated circuit (ASIC).	[A3, page 7, lines 16-21 and pages70-128]
52. (New) The method of claim 41, wherein the target buffer is assigned by processing an initial packet with the process.	[A3, page 11, lines 4-18] "Let's say a 56k NetBIOS session message is received on the INIC. The first segment will contain the NetBIOS header, which contains the total NetBIOS length. A small chunk of this first segment is provided to the host by filling in a small receive buffer, modifying the interrupt status register on the host, and raising the appropriate interrupt line. Upon receiving the interrupt, the host will read the ISR, clear it by writing back to the INIC's Interrupt Clear Register, and will then process its small receive buffer queue looking for receive buffers to be processed. Upon finding the small buffer, it will indicate the small amount of data up to the client to be processed by NetBIOS. It will also, if necessary, replenish the receive buffer pool on the INIC by passing off a pages worth of small buffers. Meanwhile, the NetBIOS client will allocate a memory pool large enough to hold the entire NetBIOS message, and will pass this address or set of addresses down to the transport driver. The transport driver will allocate an INIC command buffer, fill it in with the list of addresses, set the command type to tell the INIC that this is where to put the receive data, and then pass the command off to the INIC by writing to the command register.
53. (New) The method of claim 41, wherein the target buffer is identified with a memory descriptor list (MDL).	[A3, page 21, lines 37-47] In the "large data input" case, where "bytes available" exceeds the packet length, the TDI client will then provide an MDL, associated with an IRP, which must be completed when this MDL is filled. (This IRP/MDL may come back either in the response to TCP's call of the receive handler, or as an explicit TDI_RECEIVE request.) The ATCP driver will build a "receive request" from the MDL information, and pass this to the INIC. This request will contain: ▯ The TCP context identifier. ▯ Size and offset information. ▯ A list of physical addresses corresponding to the MDL pages."
54. (New) A method for transferring data on a network from a data source to an end station, the end station comprising a host portion and a network	[A3, page 2, lines 36-37] "A 64k SMB request (write or read-reply) is typically made up of 44 TCP segments when running over Ethernet"... "Alacritech was formed with the idea that the network processing described above could be offloaded onto a cost-effective Intelligent Network Interface Card (INIC)." [A3, page 3, lines 34-35]

interface portion, the host portion executing a multi-layer network protocol including a network layer and at least one higher layer, the data being transferred through the network interface portion of the end station, comprising:	[A3, page 6] The figure shows an end station. The figure also shows a multi-layer network protocol, including a network layer and at least one higher layer. [A3, page 6, lines 7-8] "In the fast path case, network data is given to the host after the headers have been processed and stripped." [A3, page 18, lines 22-26] "This section outlines the design specification for the Alacritech TCP (ATCP) transport driver. The ATCP driver consists of three components: 1. The bulk of the protocol stack is based on the FreeBSD TCP/IP protocol stack. This code performs the Ethernet, ARP, IP, ICMP, and (slow path) TCP processing for the driver..." [A3, page 6] The figure shows the INIC interfacing the end station to the Ethernet network.
transferring from the host portion to the network interface portion a connection definition, the connection definition comprising an IP source address, an IP destination address, a TCP source port and a TCP destination port;	[A3, page 6, lines 26-29] TCBs are initialized by the host....Once a connection has achieved a "steady-state" of operation, its associated TCB can then be turned over to the INIC, putting us into fast-path mode". [A3, page 25, line 20 through page 26, ilne 23] This section describes "passing" a context from the host to the INIC.
receiving onto the network interface portion a first packet of a multi-packet message, the multi-packet message being a message of a layer higher than the network layer, the first packet including a control field identifying the first packet, the multi-packet message having a data payload;	[A3, page 11, lines 4-5] A 56k NetBIOS session message is received onto the network interface card. The message is made up of multiple TCP "segments" (i.e., packets). A "small chunk of this first segment" is provided to the host. NetBIOS will allocates a memory pool large enough to hold the entire NetBIOS message, and will pass this address or set of addresses down to the transport driver. The transport driver will allocate an INIC command buffer, fill it in with the list of addresses, set the command type to tell the INIC that this is where to put the receive data, and then pass the command off to the INIC by writing to the command register.
reading onto the network interface portion a command from a command buffer, the command including an identifier of a target buffer, the target buffer having been assigned on the host portion by a process at a layer higher than the network layer;	[A3, page 10, line s 33-34] The INIC then reads the contents of the command buffer into its memory so that it can execute the desired command. [A3, page 14, line 41, through page 15, line 7] This command buffer will include a command buffer handle, command field, possibly a TCP context identification, and a list of physical data pointers."
determining based on the control field in the network interface portion whether the first packet corresponds to the connection definition, and if so transferring the data payload of the multi-packet message from the network interface portion directly to the target buffer; and	[A3, page 7, lines 11-21] When the packet is received, the INIC determines whether is belongs to one of its TCBs. The INIC compares the IP source and destination addresses and the source and destination TCP ports with those in each of the TCBs to determine if the packet is associates with one of its TCBs. [A3, page 8, lines 7-11] The INIC will DMA the remainder of the data into its final destination. [A3, page 8, lines 29-31] In the case of a large (56k for example) NetBIOS session message, all but the first couple hundred bytes will be DMA'd to their final destination in memory. [A3, page 11, lines 18-22] When the INIC receives the command buffer, it will DMA the remainder of the NetBIOS data, as it is received, into the memory address or addresses designated by the host."
the network interface portion writing to a response buffer and thereby indicating to the host portion completion of the command.	[A3, page 11, lines 18-22] Once the entire NetBIOS transaction is complete, the INIC will complete the command by writing to the response buffer with the appropriate status and command buffer identifier. [A3, page 16, lines 19-20] When that CPU has completed a command, it extracts the command buffer handle and passes it back to the host via a response buffer. [A3, page 10, lines 39-41] The response buffers only purpose is to indicate the completion of the designated command buffer, and to pass status about the completion.

55.(New) The method of Claim 50, wherein the network interface portion writes status information and a command buffer identifier into the response buffer.	[A3, page 11, lines 18-22] Once the entire NetBIOS transaction is complete, the INIC will complete the command by writing to the response buffer with the appropriate status and command buffer identifier.
56.(New) The method of Claim 50, wherein the identifier of the target buffer is a list of addresses.	[A3, page 11, lines 4-5] The transport driver will allocate an INIC command buffer, fill it in with the list of addresses, set the command type to tell the INIC that this is where to put the receive data, and then pass the command off to the INIC by writing to the command register.
57.(New) The method of Claim 50, wherein the identifier of the target buffer is a list of data pointers.	[A3, page 14, line 41, through page 15, line 7] This command buffer will include a command buffer handle, command field, possibly a TCP context identification, and a list of physical data pointers."
58.(New) The method of Claim 50, wherein the connection definition is a part of a Transmit Contol Block (TCB).	
59.(New) The method of Claim 50, wherein the data payload is transferred to the host portion after the headers of the packets of the multi-packet message have been stripped.	[A3, page 6, lines 7-9]

CONCLUSION

In view of the above amendments and remarks, Applicants request that an interference be declared between the present application (09/692,561) and U.S. Patent No. 6,246,683. If the Examiner would like to discuss any aspect of this application including how the claims are supported by Applicants' specification, the Examiner is requested to call the undersigned at (925) 484-9295.

Respectfully submitted,

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as Express Mail Label No.EL928365297US in an envelope addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231, on June 12, 2002.

Date: 6-12-02

Mark Lauer

Mark Lauer
Reg. No. 36,578
7041 Koll Center Plaza,
Suite 280
Pleasanton, CA 94566
Tel: (925) 484-9295
Fax: (925) 484-9291